Tracking Dense Substructures in a Dynamic Graph

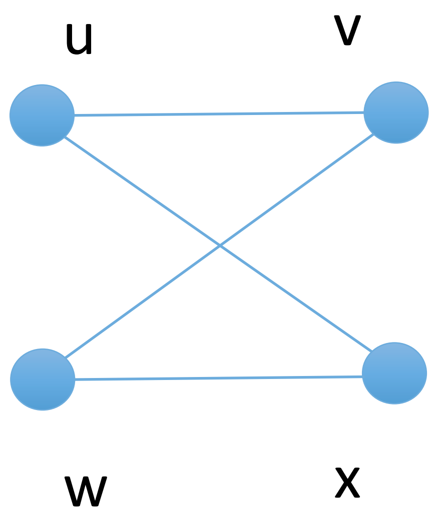
Srikanta Tirthapura

Iowa State University

(joint work with Michael Svendsen, Apurba Das, Sneha Singh)

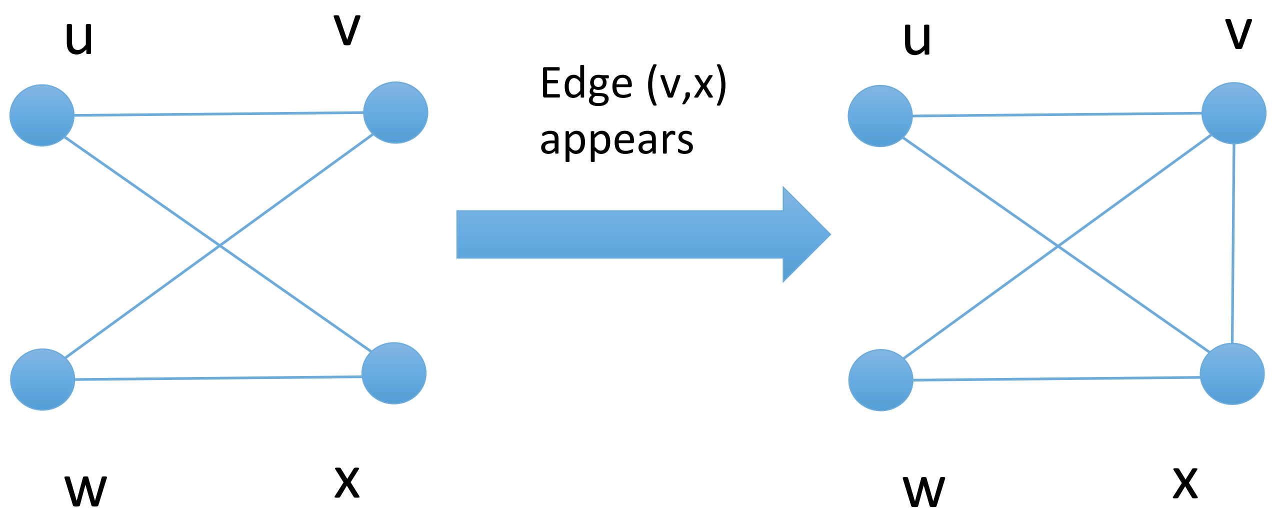
Problem

Track Emerging (and Disappearing) dense substructures in a large dynamic graph



Problem

Track Emerging (and Disappearing) dense substructures in a large dynamic graph

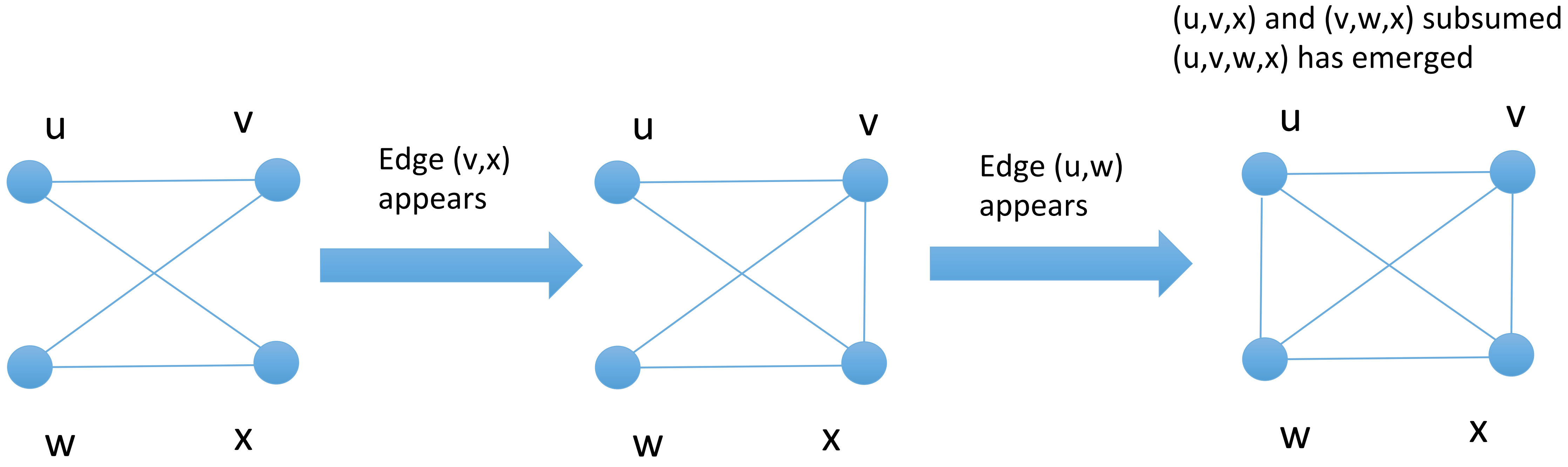


Clique (w,v,x) has emerged
Clique (u,v,x) has emerged

Cliques (v,w) (u,v) (u,x) (w,x) are subsumed by other cliques

Problem

Track Emerging (and Disappearing) dense substructures in a large dynamic graph



Why Track Dense Substructures in a Dynamic Graph?

- Key Problem in Real-Time Big Data Analytics
- Real-Time News Mining: Used in detecting emerging news stories on Twitter (Angel et al., 2012 and 2014)

Many Notions of a Dense Substructure in a Graph

- Maximal Cliques
- Maximal Bicliques
- Near-Cliques
- Near-Bicliques
- Densest Subgraph, Triangle-Densest Subgraphs
- K-Core, K-Plex
- …

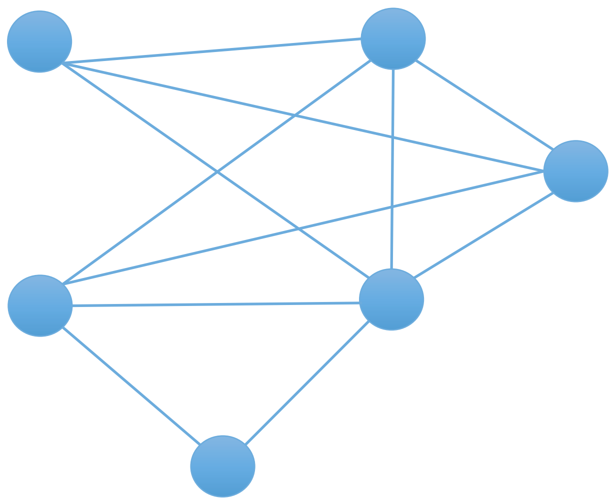
Maximal Clique

- A clique C in a graph G=(V,E) is a subset of V such that there is an edge between each pair of vertices in C
- A clique C is maximal if it is not contained within any other clique in G

Maximal Clique Enumeration Problem

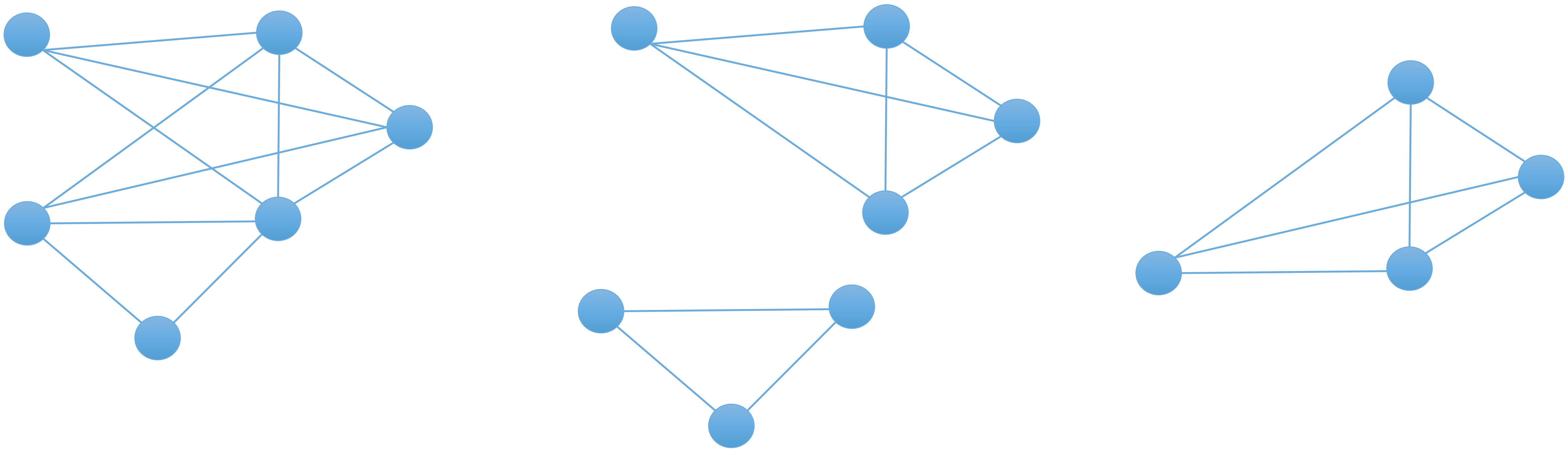
Given an undirected graph G = (V,E), enumerate all maximal cliques

Let Cliques(G) denote all maximal cliques in G



Maximal Clique Enumeration

Given an undirected graph G = (V,E), enumerate all maximal cliques in G



Dynamic Maximal Cliques

Initial graph G = (V,E)
Add a set of edges E' to get G' = (V, E + E')
C = Cliques(G), C' = Cliques(G')

Model Stores the Entire Graph
But not the set of structures in the graph

New Cliques	N(G,G') = C' – C
Deleted (Subsumed) Cliques	D(G,G') = C – C'
Symmetric Difference	S(G,G') = N U D

Questions:

- Size of Change: How large can N(G,G'), D(G,G'), S(G,G') be?
- How to Enumerate Elements of N, D, S (without enumerating C and C')?

Naïve Solution to Dynamic Maximal Cliques

- Enumerate C = Cliques(G)
- Enumerate C’ = Cliques(G + E’)

- Compute the difference

- Problems:
  - The difference maybe small while the sets are large
  - Space is an issue, since the sets of maximal cliques can be large

Questions

- Size of Change: How large are N(G,G'), D(G,G'), S(G,G')
- How to Enumerate Elements of N, D, S (without enumerating all maximal cliques in C and C')?
- Is Enumeration possible in a change-sensitive manner (time proportional to size of change)?

Maximal Cliques in a Static Graph

Moon and Moser (1965)

The largest possible size of Cliques(G) is on an n-vertex graph is f(n), where

$$
\begin{aligned}
f(n) &= 3^{n/3} && \text{if } n \bmod 3 = 0 \\
&= 4.3^{(n-4)/3} && \text{if } n \bmod 3 = 1 \\
&= 2.3^{(n-2)/3} && \text{if } n \bmod 3 = 2
\end{aligned}
$$

The above bound can be achieved by specific graphs (called Moon-Moser Graphs)

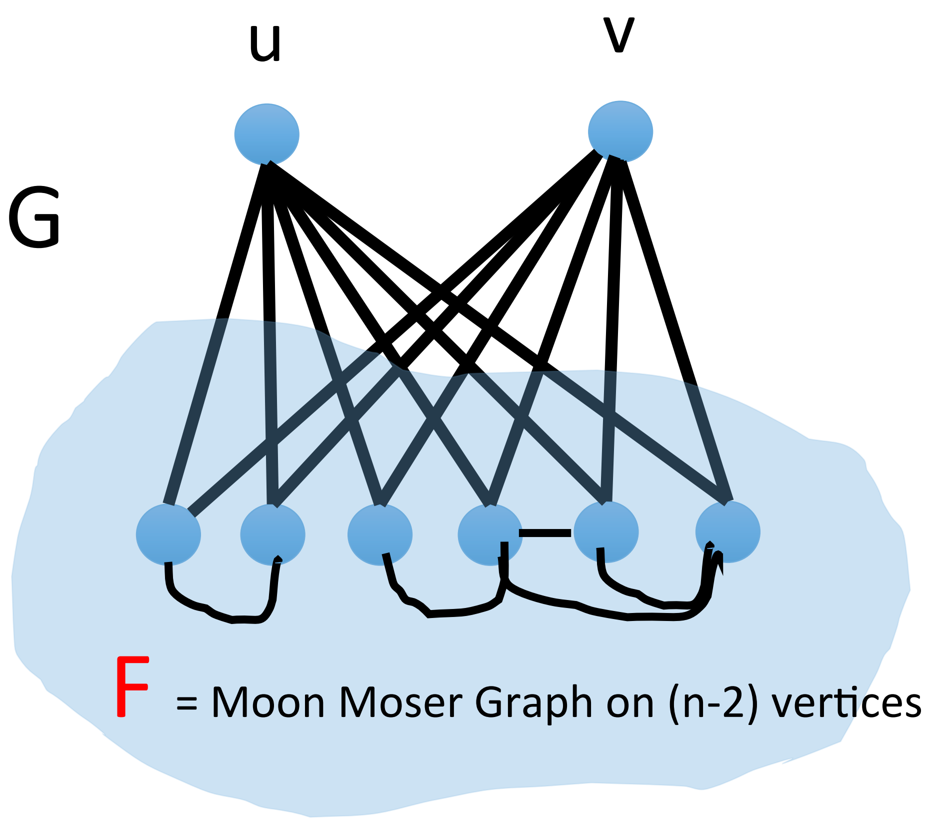
Dynamic Cliques: Single Edge Addition

Result

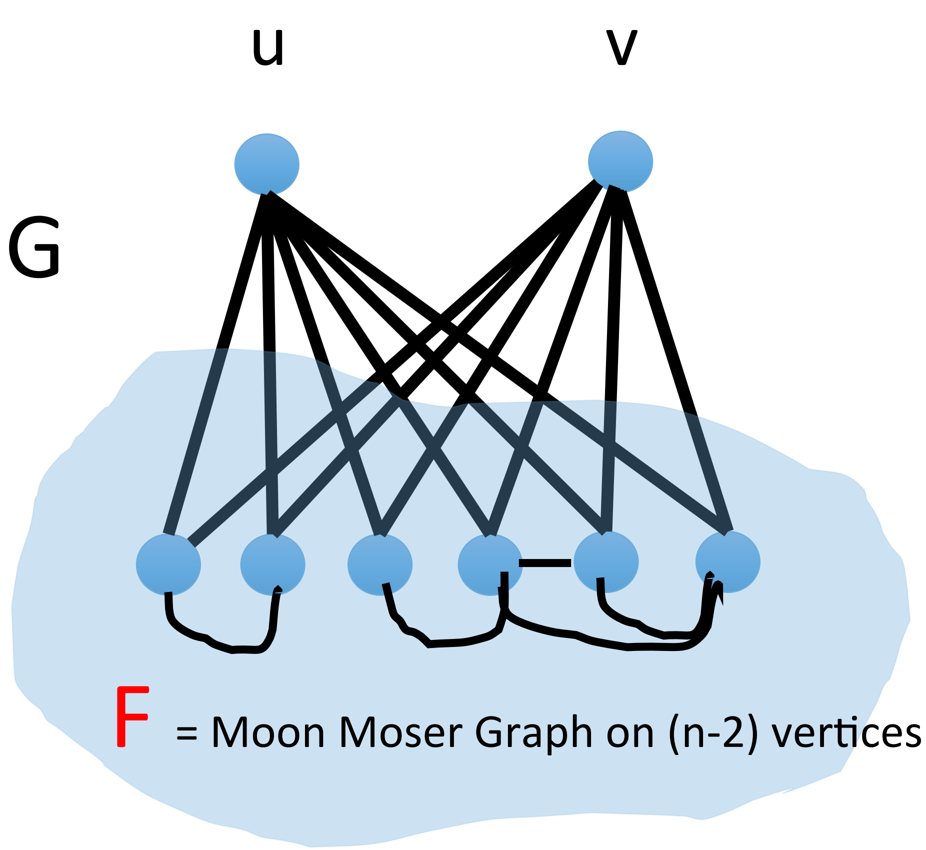
When a single edge e=(u,v) is added to a n vertex graph G, to get G'

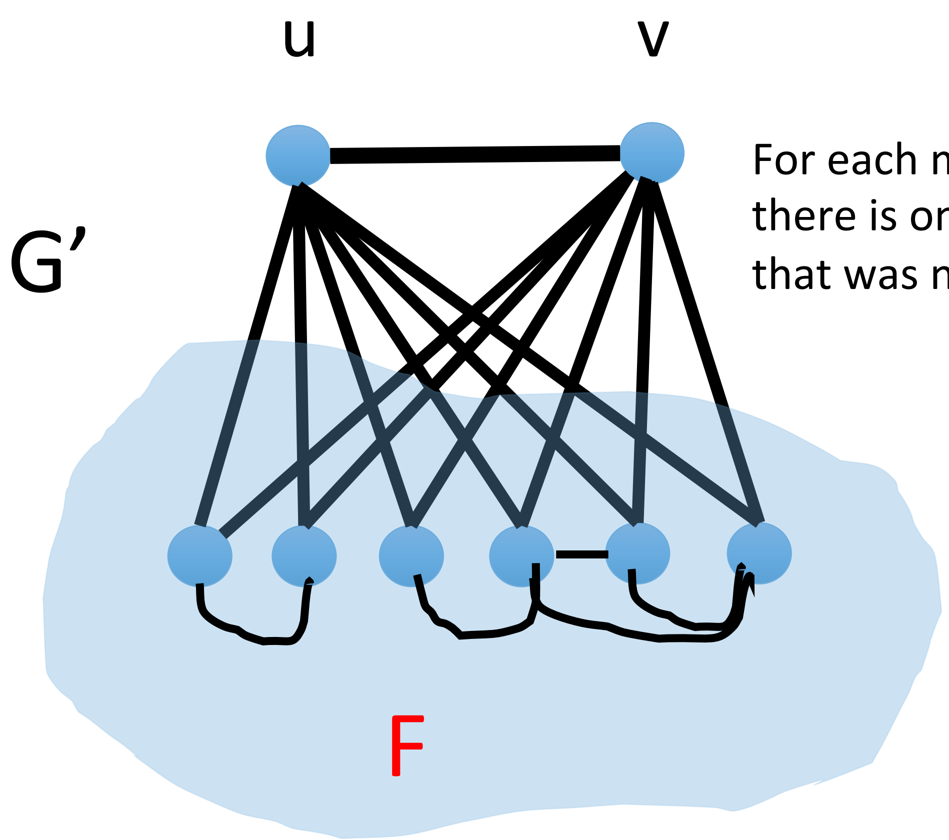
- There exist G, G' such that $|S(G,G')| = 3\ f(n-2)$
- For any G, G', $|S(G,G')| \leq 3\ f(n-2)$

Single Edge e=(u,v) added



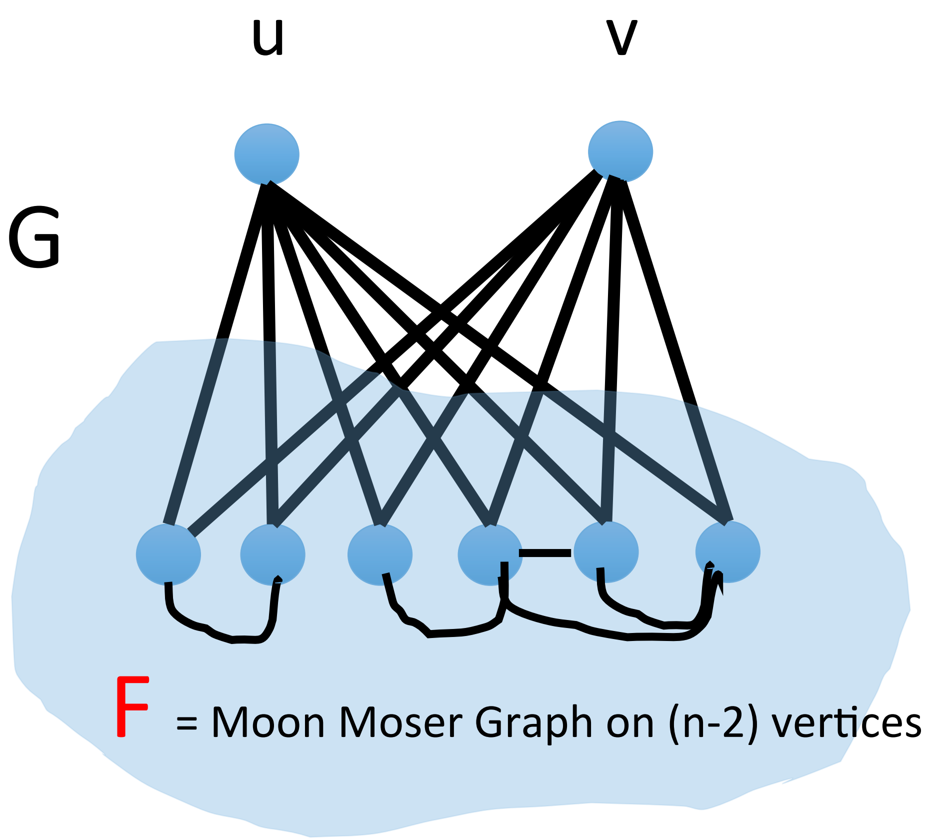
Single Edge e=(u,v) added: New Cliques



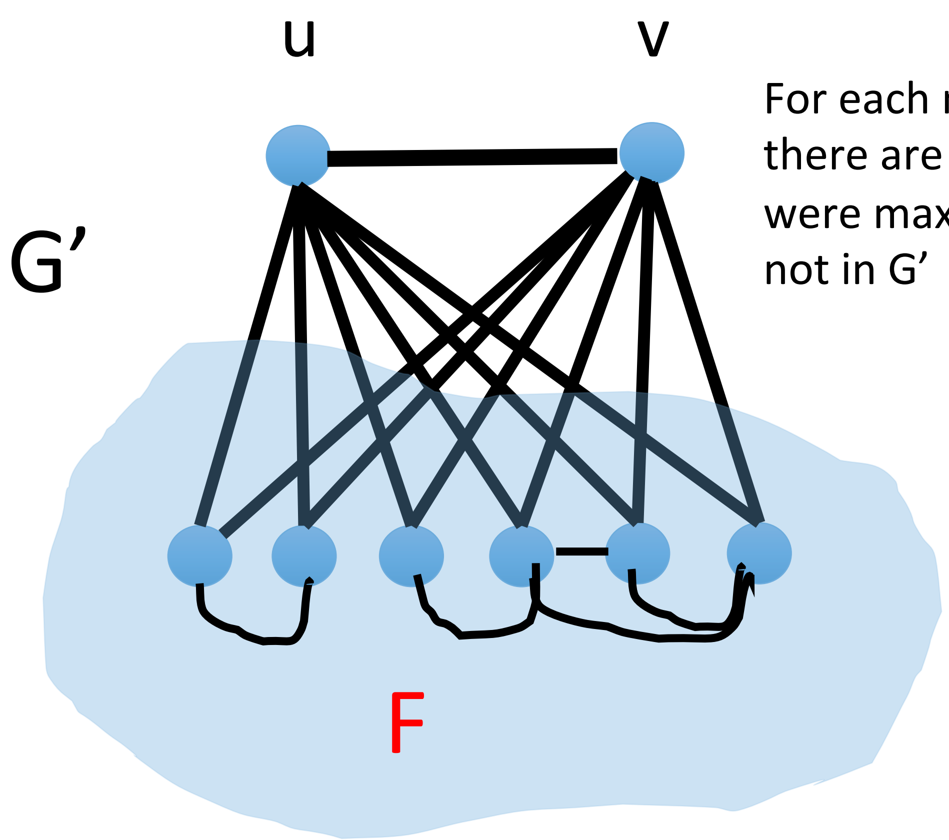


For each maximal clique in F there is one new clique in G' that was not present in G

Single Edge e=(u,v) added: Subsumed Cliques



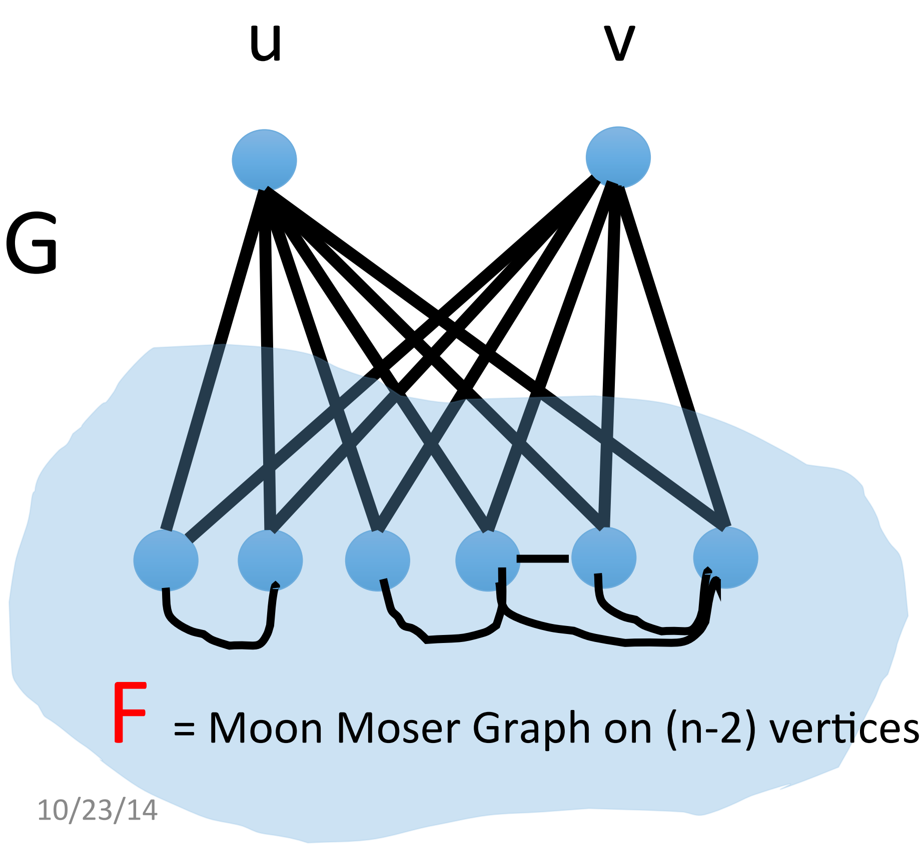
= Moon Moser Graph on (n-2) vertices

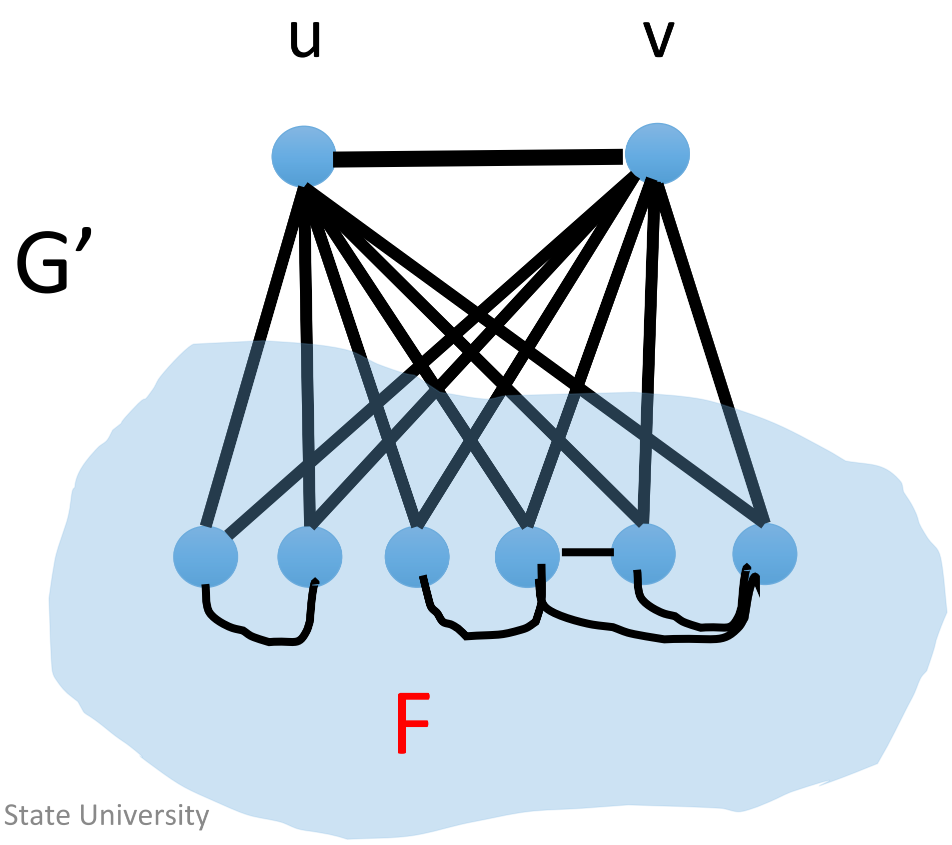


For each maximal clique in F there are two cliques that were maximal in G, but not in G’

Single Edge e=(u,v) added: Total Change

When a single edge e=(u,v) is added to a n vertex graph G, to get G'
There exist G, G' such that |S(G,G')| = 3 f(n-2)

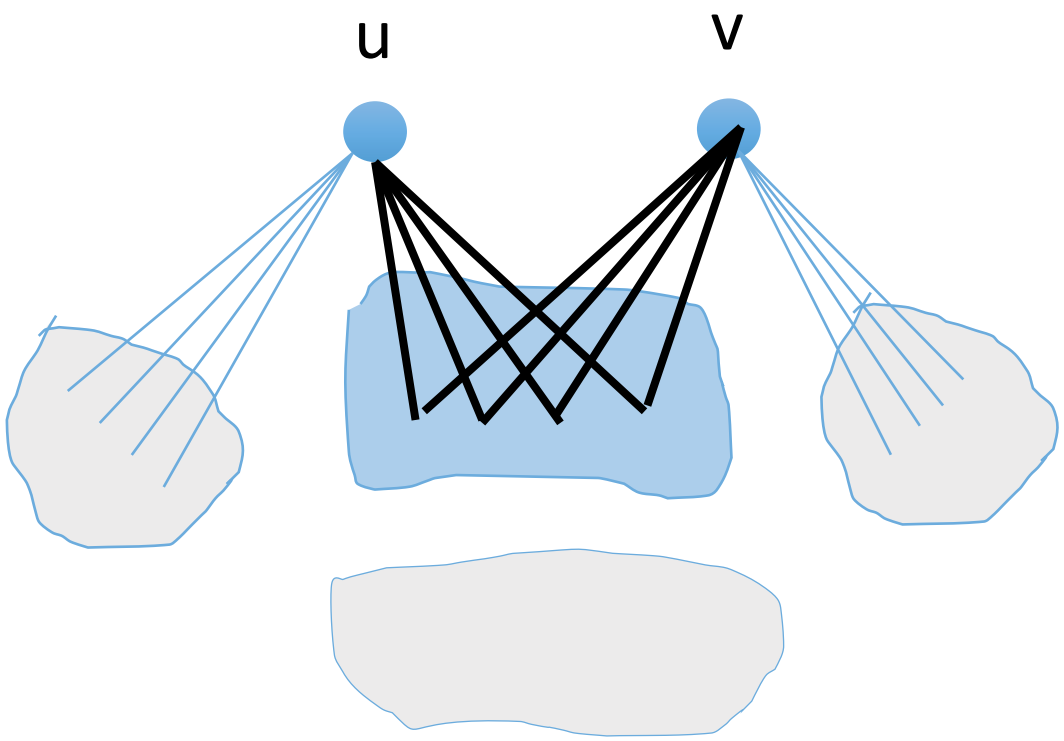




Single Edge e=(u,v) added: Total Change

When a single edge e=(u,v) is added to a n vertex graph G, to get G'

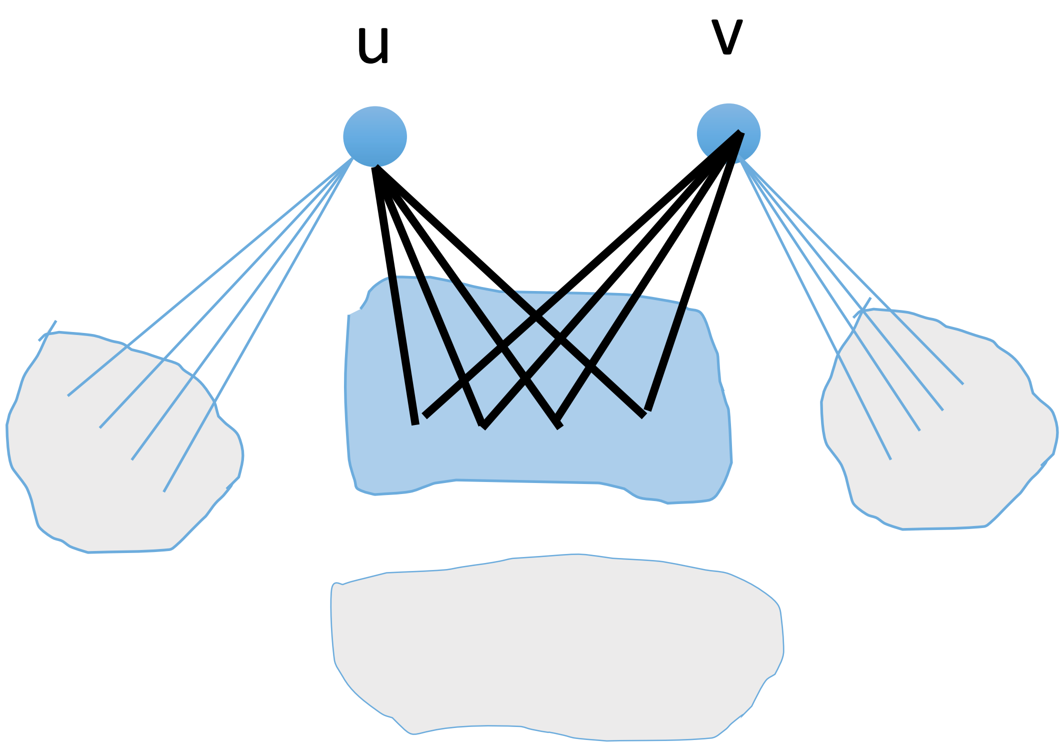
For any G, G', it must be true that $|S(G,G')| \leq 3\, f(n-2)$

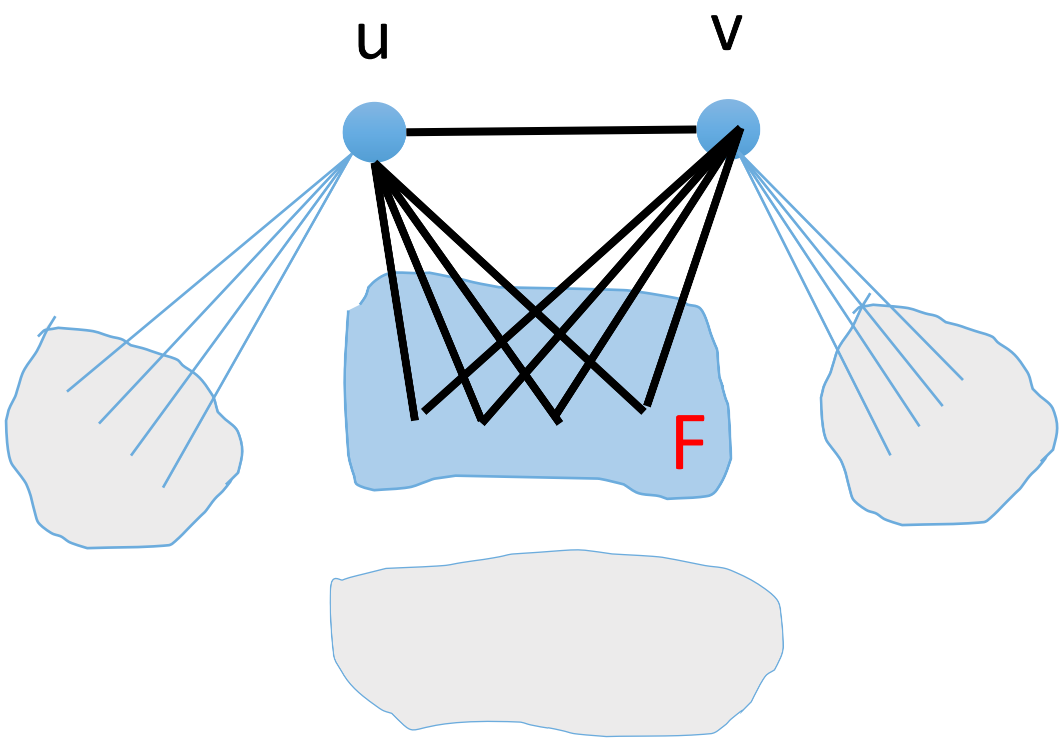


Single Edge e=(u,v) added: New Cliques

When a single edge e=(u,v) is added to a n vertex graph G, to get G'
For any G, G', it must be true that |N(G,G')|≤ f(n-2)

Each new maximal clique in G'
must have been a maximal clique in F





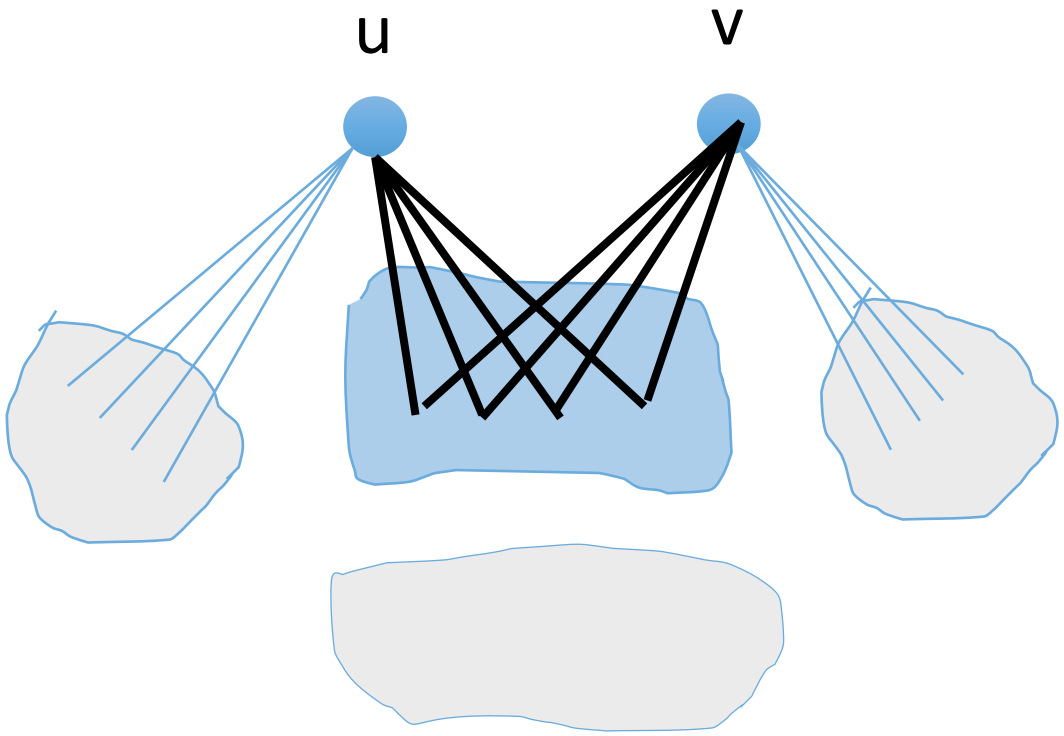
Single Edge e=(u,v) added: Subsumed Cliques

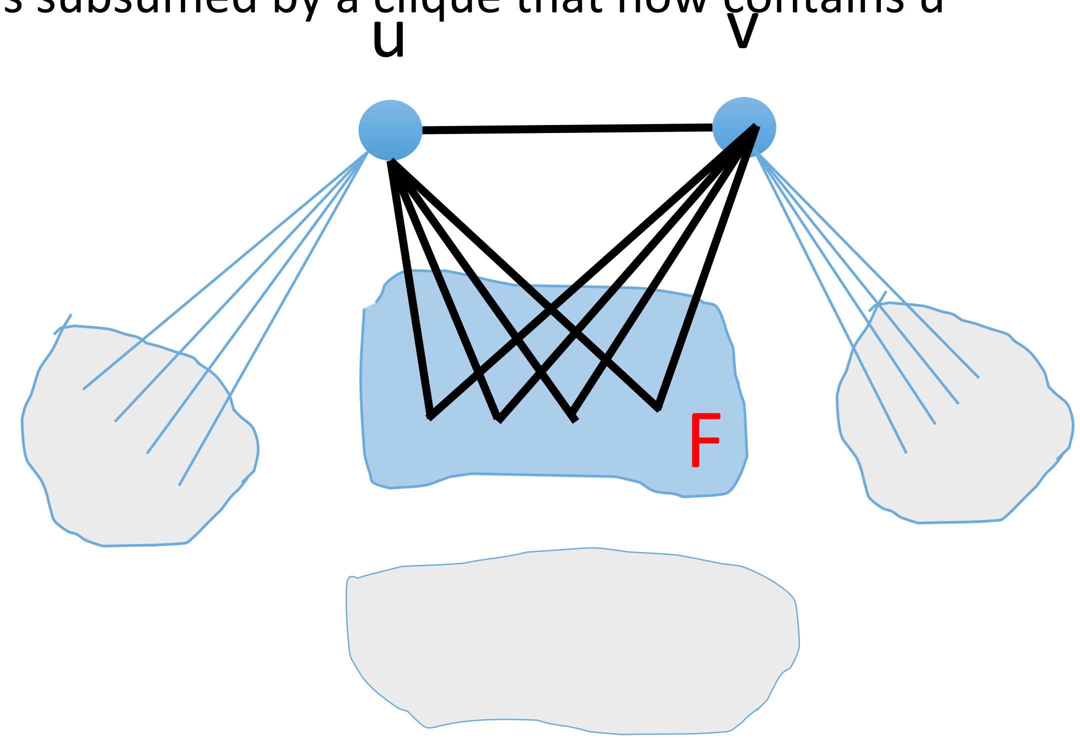
When a single edge e=(u,v) is added to a n vertex graph G, to get G’

For any G, G’, it must be true that |D(G,G’)| $\leq$ 2f(n-2)

Each subsumed maximal clique be of one of two types

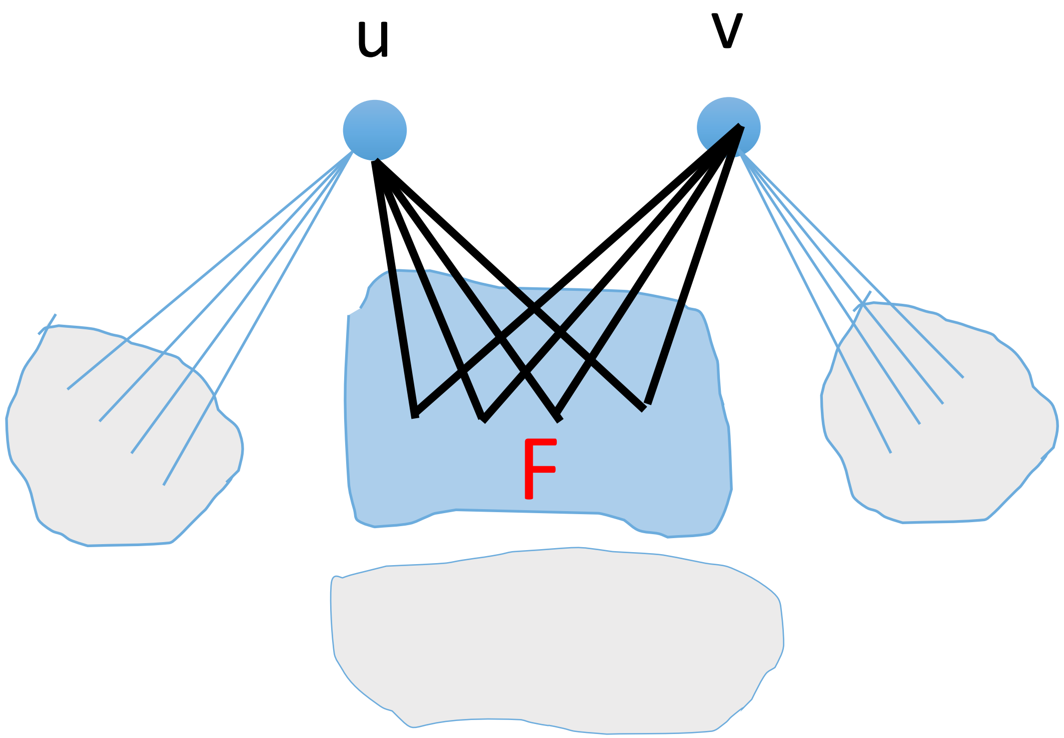
- It contains u, and is subsumed by a clique that now contains v
- It contains v, and is subsumed by a clique that now contains u





Single Edge e=(u,v) added: Enumerate Change

1. Compute $F = N(u) \cap N(v)$
2. Compute induced subgraph G(F)
3. Enumerate all maximal cliques in G(F)
   a. Worst-Case Optimal, such as Tomita-Tanaka-Takahashi (2006), or Eppstein-Loffler-Strash (2010)
   b. Output Sensitive, such as Tsukiyama et al. (1977) or Makino-Uno (2004)
4. For each clique c above, there is one new clique formed
5. For each clique c above, check two cliques and see if they are maximal in G; if so, these are subsumed cliques



Enumerate Change for One Edge: Resource Complexity

- Space: Graph G needs to be stored, but not Cliques(G)

- Time to enumerate change is proportional to the size of the change

Multiple Edges $\{e_1, e_2, \ldots e_k\}$ added

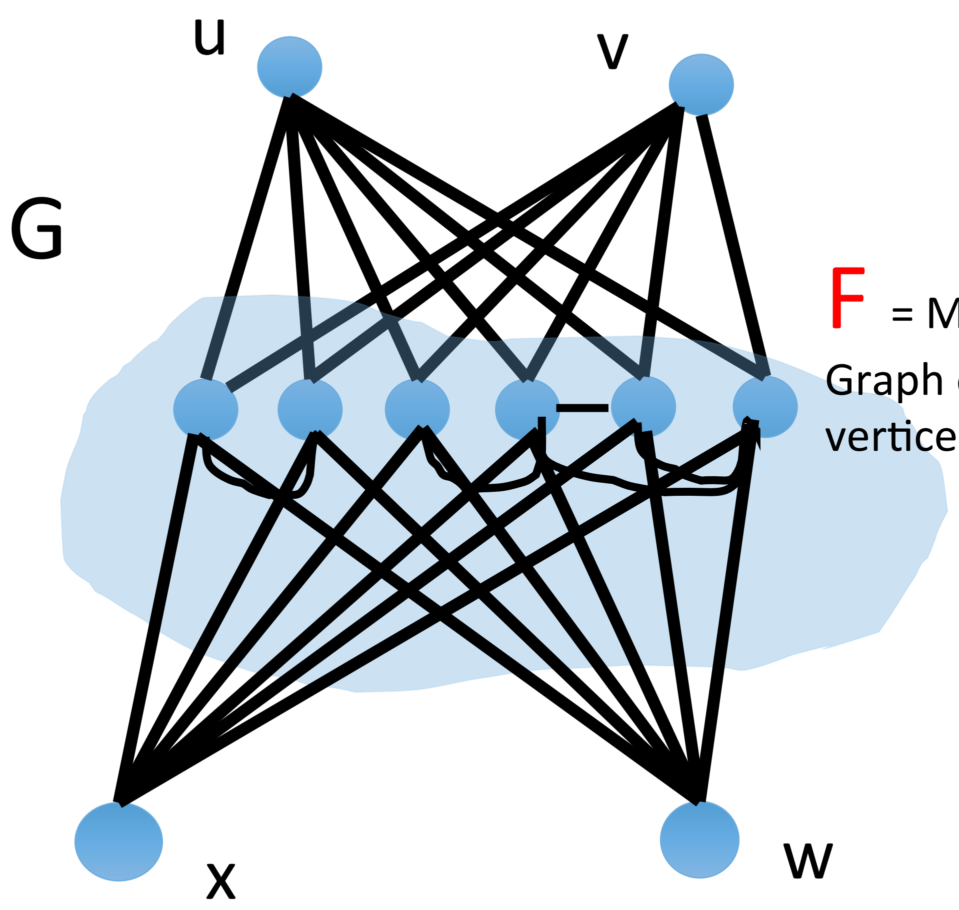
Result

When a subset of edges E' added to a n vertex graph G, to get G'

- There exist G, G' such that $|S(G,G')| = 1.849\ f(n)$

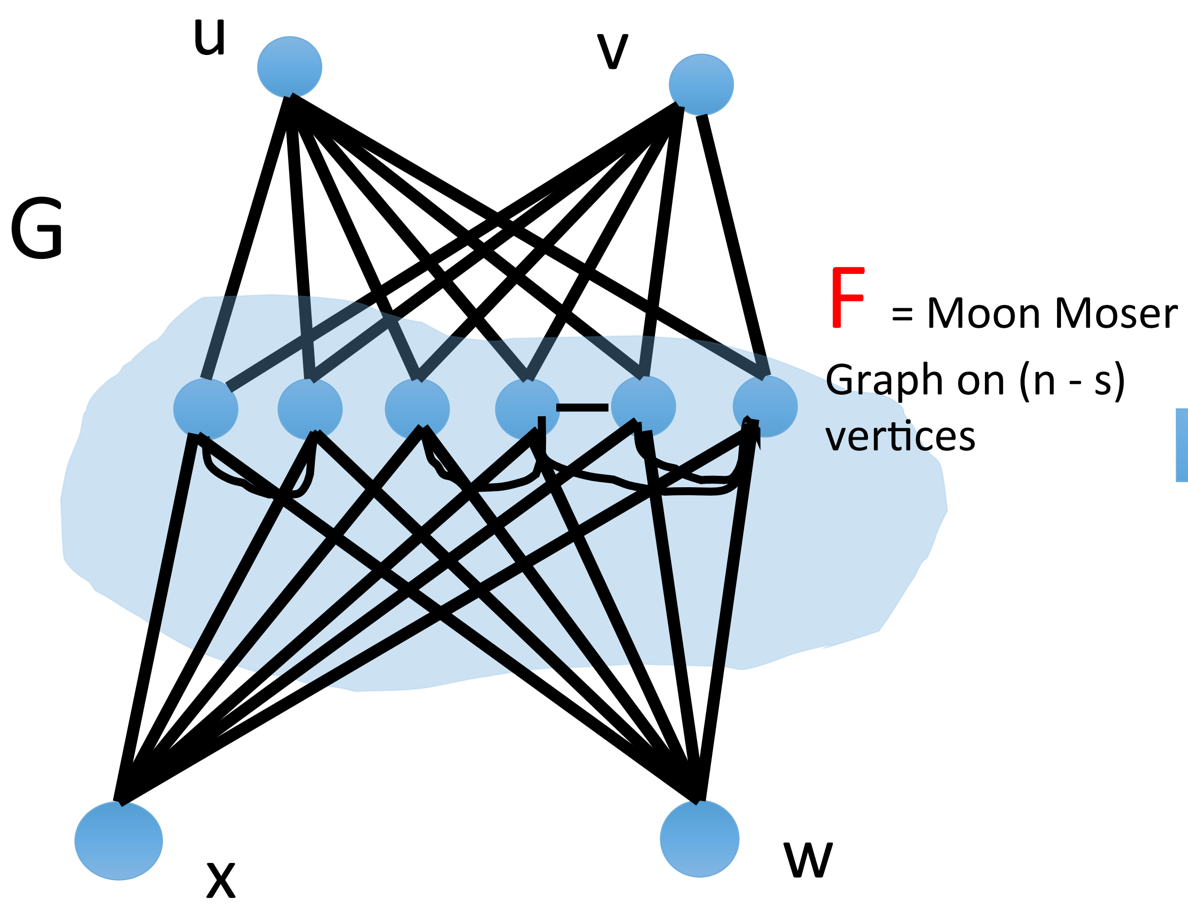
(Conjecture) For any G, G', $|S(G,G')| \leq 1.849\ f(n)$?
Note that 2f(n) is a trivial upper bound

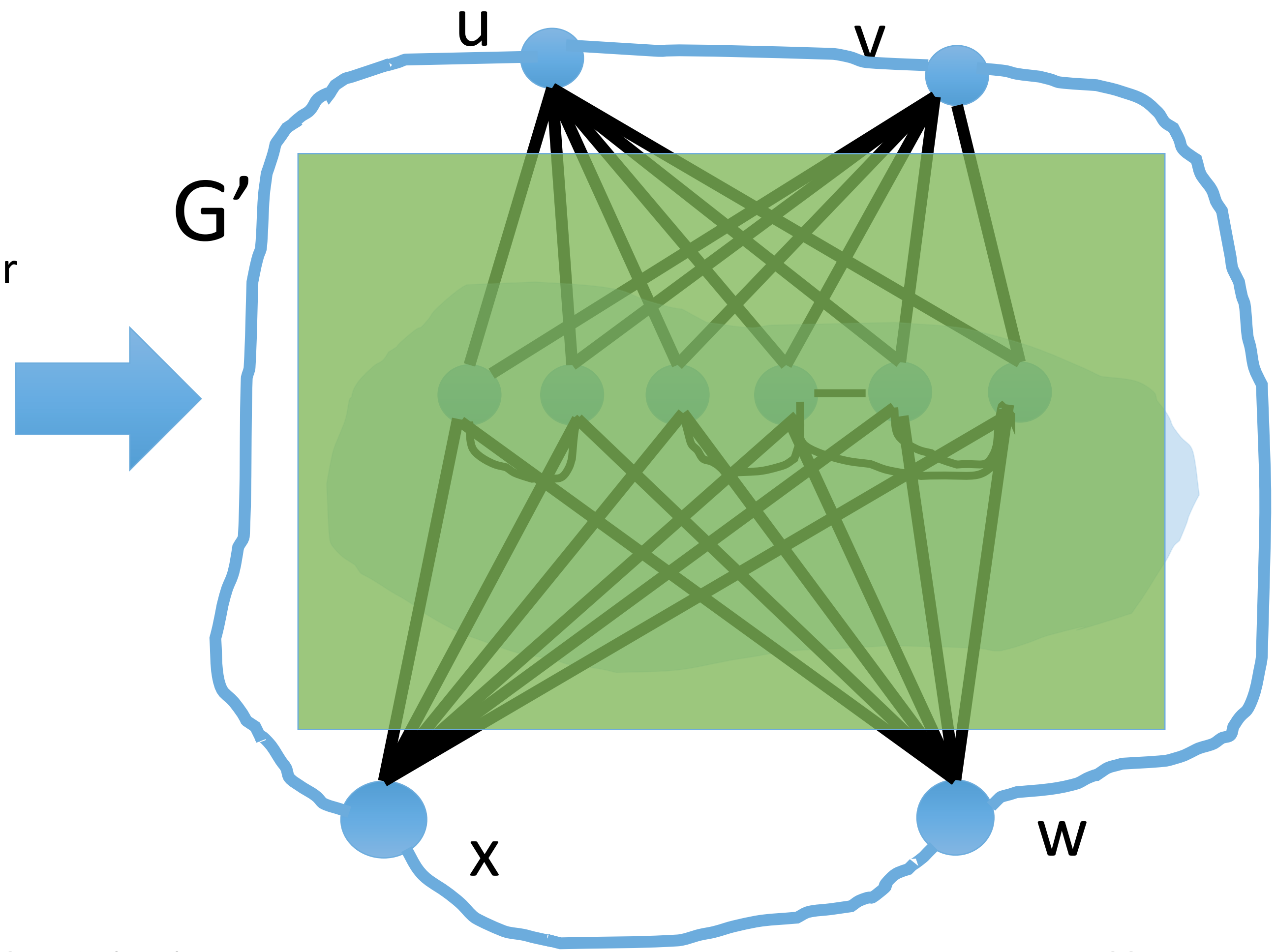
Multiple Edges added: $|S(G,G')| \geq 1.849f(n)$



F = Moon Moser Graph on (n - s) vertices

Multiple Edges added: $|S(G,G')| \geq 1.849f(n)$



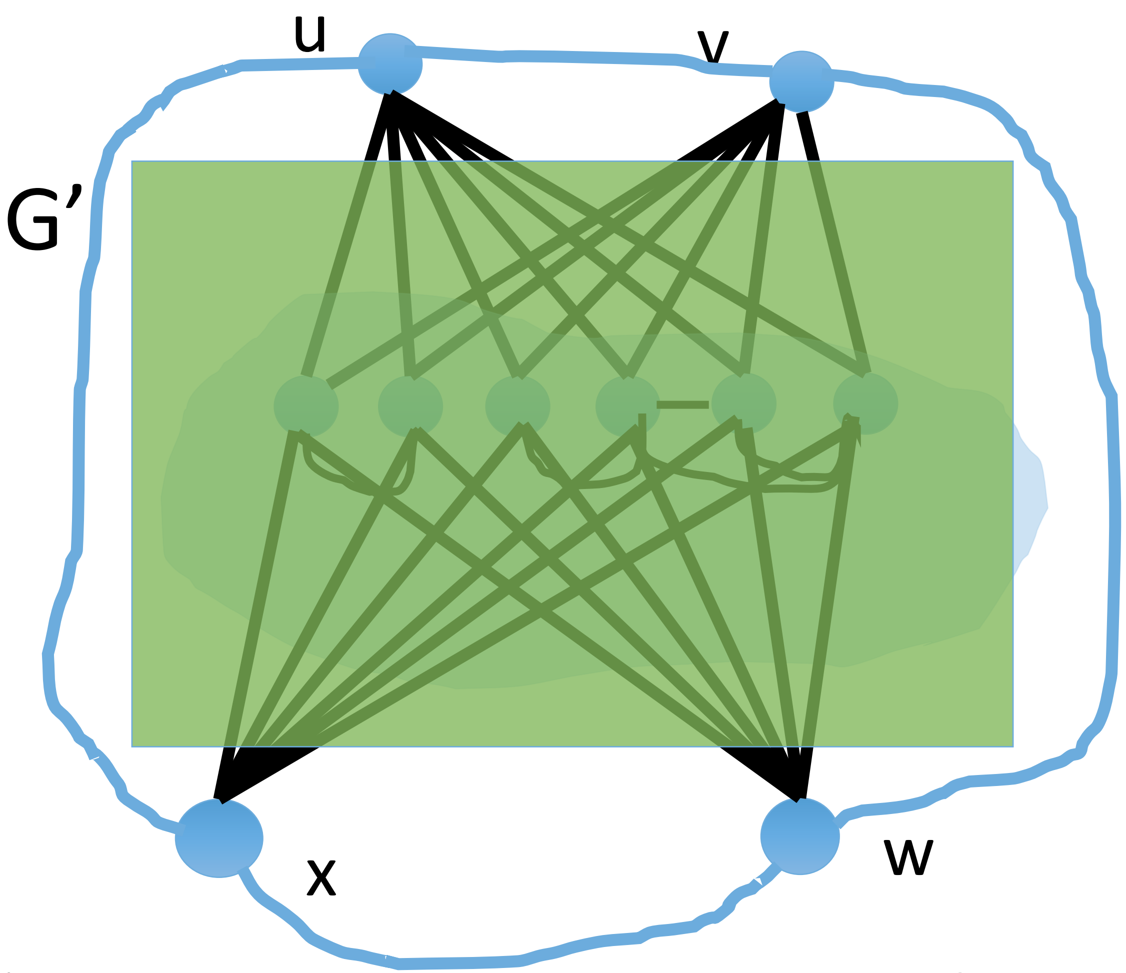


Multiple Edges added: $|S(G,G')| \geq 1.849f(n)$

Suppose s vertices in "circumference"
f(n-s) maximal cliques in center
f(s) maximal cliques in circumference

- Total new maximal cliques = f(s).f(n-s)
- Total subsumed maximal cliques is s.f(n-s)
- Total Change = (s+f(s)) f(n-s)

Maximize over all s to arrive at
1.849 f(n)



Multiple Edges $E' = \{e_1, e_2, \ldots, e_k\}$ added: Enumerate Change

- Obs 1: Each new clique must contain an edge from E'. Further, any maximal clique that contains an edge from E' is new.
    - To find new cliques, find all maximal cliques that have one or more edges from E'

- Obs 2: Each subsumed clique must contain a vertex incident to E', and must be subsumed by one of the new cliques

Multiple Edges $E' = \{e_1, e_2, \ldots, e_k\}$ added: Find New Cliques

Plan: Find all maximal cliques that have one or more edges from E'

Suppose edges of E' are ordered $\{e_1, e_2, \ldots, e_k\}$

1. G' = G + E'
2. For each e in E'
   - Enumerate all cliques in G' containing e
   - Output a clique c if e is the lowest edge in c among E'

Good: The algorithm is still change sensitive

Bad: Each new clique is enumerated multiple times, once for every edge that it contains from E'

Find New Cliques: Better Algorithm that Avoids Enumerating Duplicate Cliques

Based on the Algorithm of Tomita et al.

- Uses Branch and Bound
- Consider edges in order e_1, e_2, e_3, …, e_k
- When considering $e_{i,}$ enumerate only those maximal cliques that exclude edges e_1, e_2, …, e_{i-1}

Find New Cliques with Multiple Edge Addition: Resource Usage

- Space: As large as the size of the graph

- Time:
  - Possible to get time proportional to the set of new maximal cliques (using Tsukiyama et al.), with high multiplicative factors

  - Tomita et al. based algorithm is faster, but does not have above theoretical property

- For the first time, we are able to prove that time of enumeration is proportional to magnitude of change

Multiple edges E’ added to G: Find Subsumed Cliques

- Strategy: For each new clique c that is found, find all cliques that have been subsumed by c
- Q: Given a newly emerged clique C, which old maximal cliques have been subsumed by C?
  - Find all maximal cliques in C – E’
  - Test each one for maximality within G, and output if found to be maximal

Multiple edges E' added to G: Find Subsumed Cliques – Resource Usage

- Space: High, if it is necessary to avoid duplicates
- Time: Change-Sensitive with a constant exponential in |E'|
  - The number of maximal cliques that need to be checked for each emerging clique can be exponential in |E'|
  - A clique may be examined, but turn out to not be maximal in G

Summary of Results on Dynamic Maximal Clique Maintenance

	Magnitude of Change	Enumerate New Cliques	Enumerate Subsumed Cliques
Add a Single Edge	3 f(n-2)	Change-Sensitive Algorithm	Change-Sensitive Algorithm
Add Small Number of Edges	$1.849\ f(n) \leq \text{Max} \leq 2f(n)$	Change-Sensitive Algorithm	Change-Sensitive Algorithm
Add Large Number of Edges		Change-Sensitive Algorithm	??

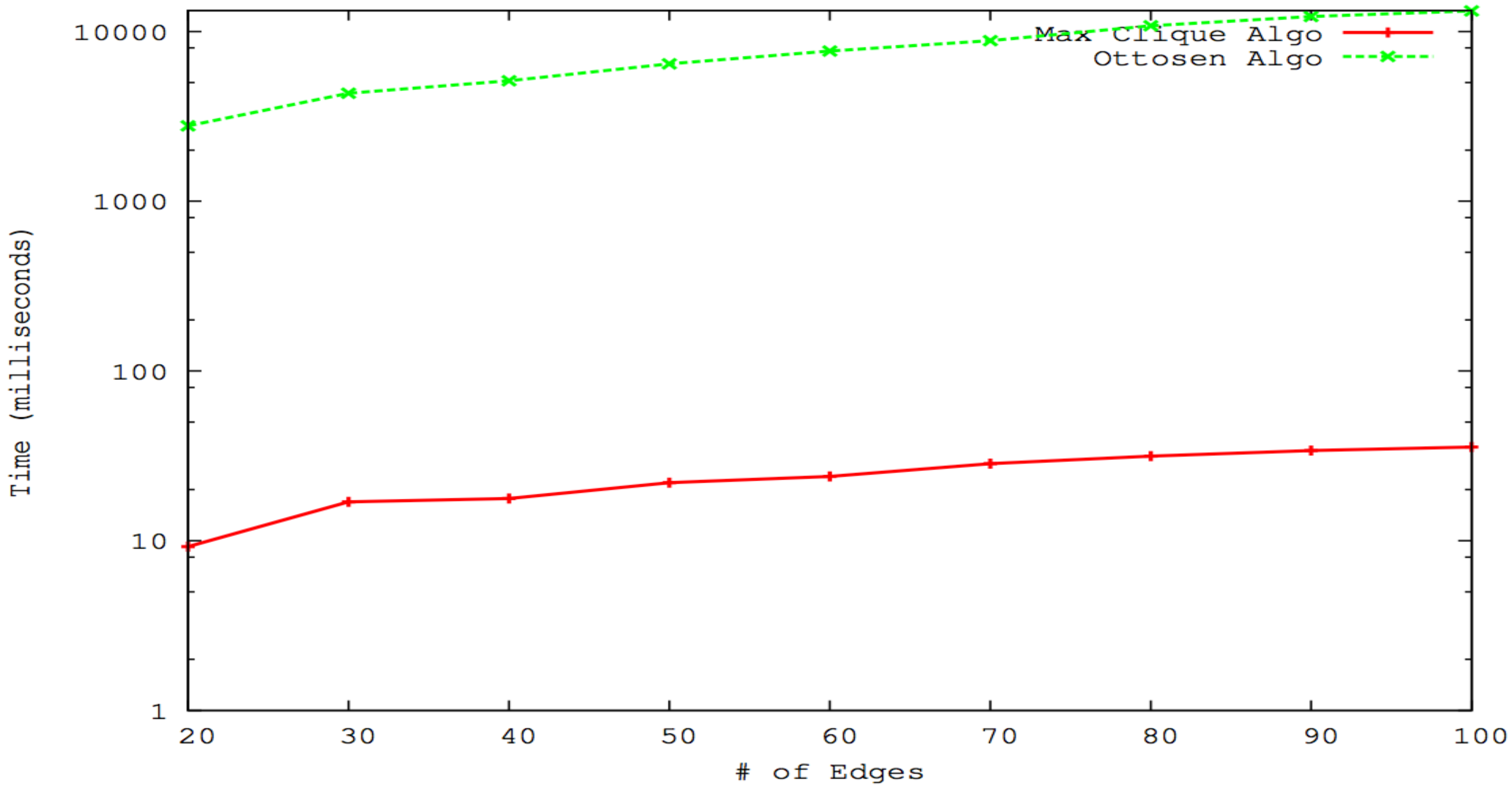
Experimental Results: Datasets Used

Graph Name	**Type Of Graph**	**# of edges**	**# of vertices**
p2p-Gnutella	Social Networks	147,892	62,586
wikivote	Wikipedia Voting Network	103,689	7,115
email-Enron	Communication Networks	367,662	36,692

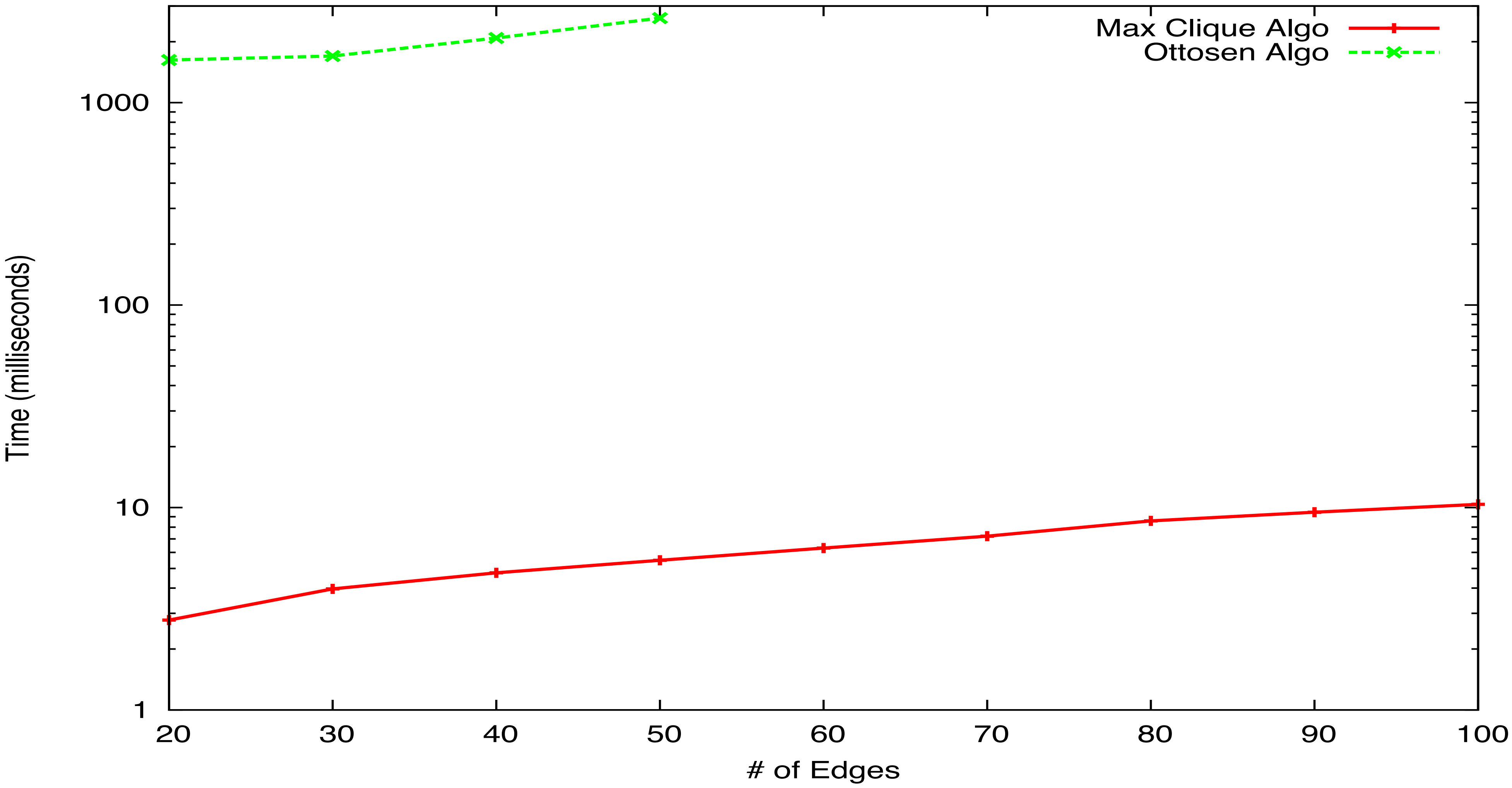
Algorithms Compared

- Maximal Clique Algorithm (Our algorithm)
- Ottosen and Vomlel "*Honor Thy Neighbor – Clique Maintenance in Dynamic Graphs*", 2010
- Stix, "*Finding All Cliques in Dynamic Graphs*", 2004
  - Took more than two hours for each data set used, hence not shown

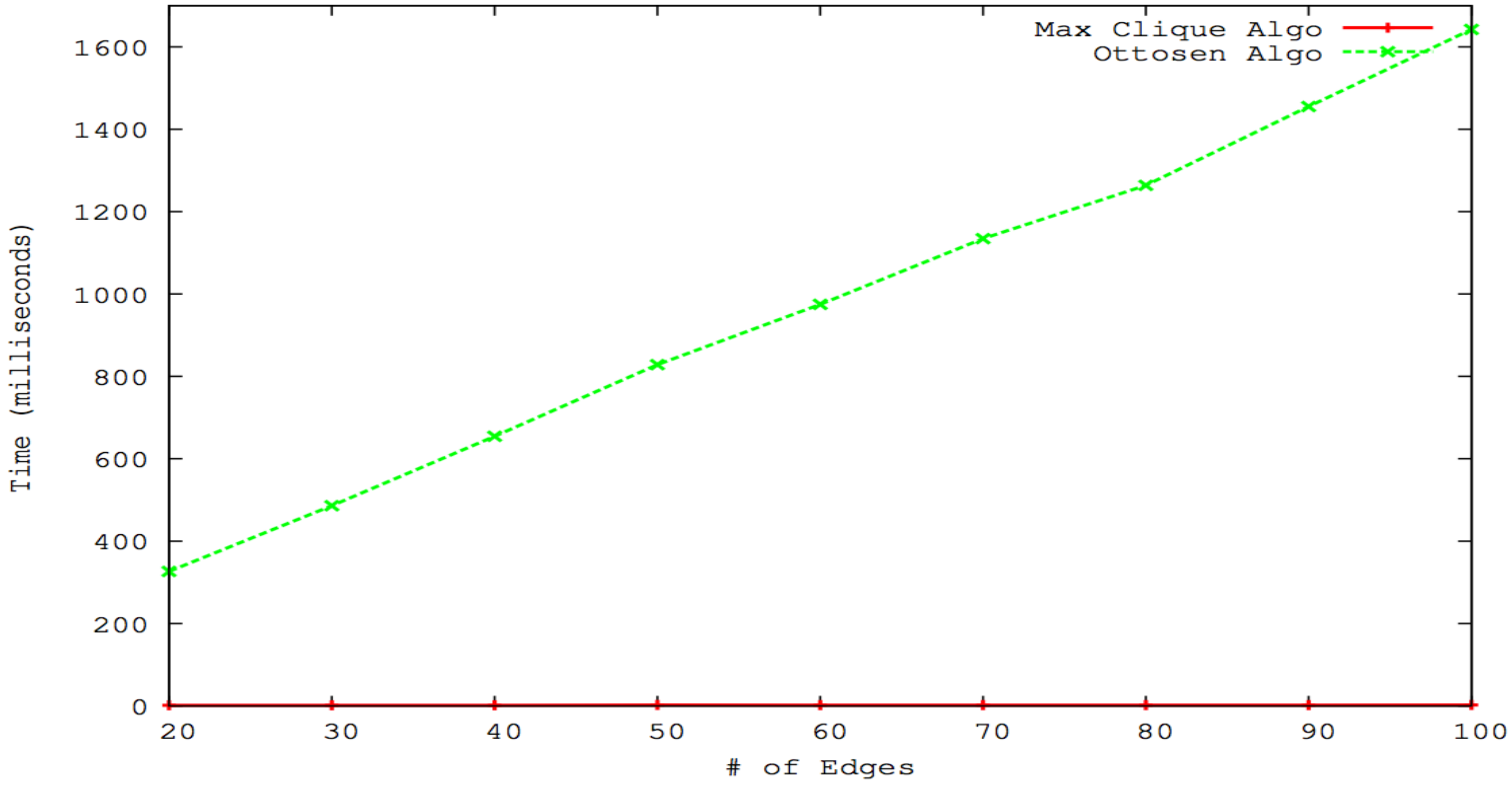
of Edges vs Time (wikivote)



of Edges vs Time (email_Enron)



of Edges vs Time (p2p-Gnutella)



Conclusion

- Systematic Exploration of Maximal Clique Maintenance in Dynamic Graphs

- Orders of magnitude speedup compared to prior work

- Open Questions
  - Tight Bounds for magnitude of Change
  - Better Method for Enumerating Subsumed Cliques
  - More Usable Characterization of Subsumed Cliques

Conclusion: Future Work

- Impose a model on the graph arrival, make use of this model
- Scaling to Very Large Graphs, Sublinear space
- Parallel Processing
- Other Dense Substructures
  - Maximal Bicliques
  - Enumerate Emergence of Only Large Structures
  - Tracking of Incomplete but Dense Structures
  - …
- Scope of Data – Sliding Window, etc